

[illegible]

PM
VO

```

LL          IIIII
LL          IIIII
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LLLLLLLLLLL
LLLLLLLLLLL

SSSSSSSSS
SSSSSSSSS
SS
SS
SS
SS
    SSSSSS
    SSSSSS
                SS
                SS
                SS
                SS
SSSSSSSSS
SSSSSSSSS

```

| | | |
|------|-----|--|
| (1) | 2 | COPYRIGHT NOTICE |
| (1) | 29 | PROGRAM DESCRIPTION |
| (2) | 56 | DECLARATIONS |
| (3) | 71 | STORAGE DEFINITIONS |
| (4) | 106 | READ-ONLY DATA DEFINITIONS |
| (5) | 177 | INIT PFN -- INITIALIZE FOR EXAMINING PFN DATA BASE |
| (6) | 211 | DISPCAY PFN -- DISPLAY MEMORY MANAGEMENT DATA |
| (7) | 297 | SHOW PFN LIST, DISPLAY PFN LIST |
| (8) | 361 | PFN TITLE, DISPLAY PFN HEADING LINE |
| (9) | 388 | SHOW PFN, SHOW DATA ON A SINGLE PFN ENTRY |
| (10) | 465 | DISPCAY_SPT_RANGE -- DISPLAY SYSTEM PAGE TABLE W/RANGE |
| (10) | 509 | DISPLAY_SPT -- DISPLAY SYSTEM PAGE TABLE |
| (11) | 583 | DUMP PTE -- FORMAT THE PAGE TABLE |
| (12) | 813 | PTE_STATE -- SET STATE OF PTE DISPLAY |

```
0000 1 .TITLE MMG PAGE TABLE FORMATTING ROUTINES
0000 2 .SBTTL COPYRIGHT NOTICE
0000 3 .IDENT 'V04-000'
0000 4 :
0000 5 :*****
0000 6 :*
0000 7 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :* ALL RIGHTS RESERVED.
0000 10 :*
0000 11 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :* TRANSFERRED.
0000 17 :*
0000 18 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :* CORPORATION.
0000 21 :*
0000 22 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :*
0000 25 :*
0000 26 :*****
0000 27 :
```



```
0000 29 .SBTTL PROGRAM DESCRIPTION
0000 30 :++
0000 31 FACILITY
0000 32 :
0000 33 SYSTEM DUMP ANALYZER
0000 34 :
0000 35 ABSTRACT
0000 36 :
0000 37 THIS MODULE CONTAINS ROUTINES RELATING TO FORMATTED
0000 38 A SPECIFIED PAGE TABLE.
0000 39 :
0000 40 ENVIRONMENT
0000 41 :
0000 42 NATIVE MODE, USER MODE
0000 43 :
0000 44 AUTHOR
0000 45 :
0000 46 TIM HALVORSEN, JULY 1978
0000 47 :
0000 48 MODIFIED BY
0000 49 :
0000 50 V03-001 WMC0001 Wayne Cardoza 19-Aug-1982
0000 51 Correct the use of MAXPFN to allow the last page.
0000 52 :
0000 53 :--
```

| | | | | |
|------|----|---------|--------------|--------------------------------|
| 0000 | 56 | .SBTTL | DECLARATIONS | |
| 0000 | 57 | : | | |
| 0000 | 58 | : | | |
| 0000 | 59 | : | | |
| 0000 | 60 | | | |
| 0000 | 61 | SDMPDEF | | : Dump file definitions |
| 0000 | 62 | SOPDEF | | : Define opcode equivalences |
| 0000 | 63 | SOPTDEF | | : Options definitions |
| 0000 | 64 | SPFNDEF | | : Page frame data definitions |
| 0000 | 65 | SPHDDEF | | : Process header definitions |
| 0000 | 66 | SPTDEF | | : Page table entry definitions |
| 0000 | 67 | STPADEF | | : TPARSE definitions |
| 0000 | 67 | SVADEF | | : Virtual address definitions |
| 0000 | 68 | SWSLDEF | | : Working set list definitions |

```
0000 71 .SBTTL STORAGE DEFINITIONS
0000 72 :
0000 73 : WRITABLE STORAGE DEFINITIONS
0000 74 :
0000 75 :
00000000 76 .PSECT SDADATA,NOEXE,WRT
0000 77
00000040 0000 78 BUFFER:
0000 79 .BLKL 16 ; GETMEM WORK BUFFER
0040 80
00000044 0040 81 SD$GL_MAXPFN:
00000042 0040 82 .BLKL 1 ; VALUE OF MMG$GL_MAXPFN
00000042 0044 83 MMG$GW_BIGPFN = SD$GL_MAXPFN + 2
00000048 0044 84 SD$AB_STATE:
00000048 0044 85 .BLKL 1 ; VALUE OF PFNSAB_STATE
0000004C 0048 86 SD$AB_TYPE:
0000004C 004C 87 .BLKL 1 ; PFNSAB_TYPE
00000050 004C 88 SD$AW_REFCNT:
00000050 0050 89 .BLKL 1 ; PFNSAW_REFCNT
00000054 0050 90 SD$AL_BAK:
00000054 0054 91 .BLKL 1 ; PFNSAL_BAK
00000058 0054 92 SD$AL_PTE:
00000058 0058 93 .BLKL 1 ; PFNSAL_PTE
0000005C 0058 94 SD$Ax_FLINK:
0000005C 005C 95 .BLKL 1 ; PFNSAW_FLINK
00000060 005C 96 SD$Ax_BLINK:
00000060 0060 97 .BLKL 1 ; PFNSAW_BLINK
00000064 0060 98 SD$Ax_WSLX:
00000064 0064 99 .BLKL 1 ; WORKING SET INDEX
00000000 100
00000000 101 .PSECT MMG,EXE,NOWRT
0000 102
0000 103 .DEFAULT DISPLACEMENT, LONG
```



```
0000 106 .SBTTL READ-ONLY DATA DEFINITIONS
0000 107
0000 108 :
0000 109 : READ-ONLY DATA DEFINITIONS
0000 110 :
0000 111
0000 112 PTECTL1:
0000 113 STRING < !_!XL !XL !XL !AD !AD !AD !AD !AD >
002F 114 PTECTL2_WORD:
002F 115 STRING < !_!XL !XL !XL !AD !AD !AD !AD !AD !AD !AD !XB !XB !6UW !X
008B 116 PTECTL2_LONG:
008B 117 STRING < !_!XL !XL !XL !AD !AD !AD !AD !AD !AD !AD !XB !XB !6UW !X
00E7 118
00E7 119 PROT_TABLE:
45 4E 4F 4E 00E7 120 .ASCII /NONE/
2A 2A 2A 2A 00EB 121 .ASCII /****/
20 20 57 4B 00EF 122 .ASCII /KW /
20 20 52 4B 00F3 123 .ASCII /KR /
20 20 57 55 00F7 124 .ASCII /UW /
20 20 57 45 00FB 125 .ASCII /EW /
57 4B 52 45 00FF 126 .ASCII /ERKW/
20 20 52 45 0103 127 .ASCII /ER /
20 20 57 53 0107 128 .ASCII /SW /
57 45 52 53 010B 129 .ASCII /SREW/
57 4B 52 53 010F 130 .ASCII /SRKW/
20 20 52 53 0113 131 .ASCII /SR /
57 53 52 55 0117 132 .ASCII /URSW/
57 45 52 55 011B 133 .ASCII /UREW/
57 4B 52 55 011F 134 .ASCII /URKW/
20 20 52 55 0123 135 .ASCII /UR /
0127 136
0127 137 OWNER_TABLE:
55 53 45 4B 0127 138 .ASCII /KESU/
012B 139
012B 140 TYPE_TABLE:
53 4E 41 52 54 012B 141 .ASCII /TRANS/
20 58 54 50 47 0130 142 .ASCII /GPTX /
4C 49 46 47 50 0135 143 .ASCII /PGFIL/
20 20 58 54 53 013A 144 .ASCII /STX /
4F 52 45 5A 44 013F 145 .ASCII /DZERO/
44 49 4C 41 56 0144 146 .ASCII /VALID/
47 41 50 4F 49 0149 147 .ASCII /IOPAG/
20 20 20 20 20 014E 148 .ASCII / /
0153 149
0153 150 LOC_TABLE:
20 54 53 4C 45 45 52 46 0153 151 .ASCII /FREELST /
20 54 53 4C 59 46 44 4D 015B 152 .ASCII /MDFYLIST /
20 54 53 49 4C 44 41 42 0163 153 .ASCII /BADLIST /
20 44 4E 45 50 4C 45 52 016B 154 .ASCII /RELPEND /
20 52 4F 52 52 45 44 52 0173 155 .ASCII /RDERROR /
20 54 55 4F 45 47 41 50 017B 156 .ASCII /PAGEOUT /
20 20 4E 49 45 47 41 50 0183 157 .ASCII /PAGEIN /
20 20 45 56 49 54 43 41 018B 158 .ASCII /ACTIVE /
0193 159
0193 160 PAGTYP_TABLE:
20 53 53 45 43 4F 52 50 0193 161 .ASCII /PROCESS /
20 20 4D 45 54 53 59 53 019B 162 .ASCII /SYSTEM /
```


| | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|------|-----|---------------|---------|---|
| 20 | 20 | 4C | 41 | 42 | 4F | 4C | 47 | 01A3 | 163 | .ASCII | /GLOBAL | / |
| 20 | 20 | 54 | 52 | 57 | 4C | 42 | 47 | 01A8 | 164 | .ASCII | /GBLWRT | / |
| 20 | 20 | 4C | 42 | 54 | 47 | 50 | 50 | 01B3 | 165 | .ASCII | /PPGTBL | / |
| 20 | 20 | 4C | 42 | 54 | 47 | 50 | 47 | 01BB | 166 | .ASCII | /GPGTBL | / |
| 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 01C3 | 167 | .ASCII | / | / |
| 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 01CB | 168 | .ASCII | / | / |
| | | | | | | | | 01D3 | 169 | | | |
| | | | | | | | | 01D3 | 170 | MODIFY_TABLE: | | |
| | | | | | | 4D | 20 | 01D3 | 171 | .ASCII | / M/ | |
| | | | | | | | | 01D5 | 172 | | | |
| | | | | | | | | 01D5 | 173 | WSLOCK_TABLE: | | |
| | | | | | | 4C | 20 | 01D5 | 174 | .ASCII | / L/ | |

```

01D7 177 .SBTTL INIT_PFN -- INITIALIZE FOR EXAMINING PFN DATA BASE
01D7 178 :---
01D7 179 :
01D7 180 INIT_PFN
01D7 181 :
01D7 182 THIS ROUTINE MUST BE CALLED BEFORE ANY REFERENCES ARE
01D7 183 MADE TO THE PFN DATA BASE.
01D7 184 :
01D7 185 INPUTS:
01D7 186 :
01D7 187 NONE
01D7 188 :
01D7 189 OUTPUTS:
01D7 190 :
01D7 191 RO = SUCCESS FLAG
01D7 192 SDSA... CELLS ARE INITIALIZED
01D7 193 :
01D7 194 :---
01D7 195 :
0000 01D7 196 INIT_PFN::
01D7 197 .WORD 0
01D9 198
01D9 199 REQMEM @MMG$GL_MAXPFN,SDA$GL_MAXPFN
01ED 200 REQMEM @PFNSAB_STATE,SDA$AB_STATE
0201 201 REQMEM @PFNSAB_TYPE,SDA$AB_TYPE
0215 202 REQMEM @PFNSAW_REFCNT,SDA$AW_REFCNT
0229 203 REQMEM @PFNSAL_BAK,SDA$AL_BAK
023D 204 REQMEM @PFNSAL_PTE,SDA$AL_PTE
0251 205 REQMEM @PFNSAx_FLINK,SDA$Ax_FLINK
0265 206 REQMEM @PFNSAx_BLINK,SDA$Ax_BLINK
0279 207 REQMEM @PFNSAx_WSLX,SDA$Ax_WSLX
04 028D 208 RET

```

```
028E 211 .SBTTL DISPLAY_PFN DISPLAY MEMORY MANAGEMENT DATA
028E 212 ---
028E 213
028E 214 DISPLAY_PFN
028E 215
028E 216 THIS ROUTINE IS RESPONSIBLE FOR PRINTING ALL INFORMATION
028E 217 RELATING TO THE MEMORY MANAGEMENT DATA BASE.
028E 218
028E 219 INPUTS:
028E 220
028E 221 NONE
028E 222
028E 223 OUTPUTS:
028E 224
028E 225 NONE
028E 226
028E 227 ---
028E 228 .ENABL LSB
028E 229
028E 230 .ENTRY DISPLAY_PFN,^M<R2,R3,R4,R5,R6>
0290 231
0290 232 CALLS #0,INIT_PFN ; SETUP TO READ PFN DATA
38 00000000'EF 00 04 FB E1 0295 233 BBC #OPT$V_SINGLEPFN,OPTIONS,20$ ; BRANCH IF LIST WANTED
0290 234
0290 235 DISPLAY A SINGLE SPECIFIED PFN ENTRY
0290 236
0290 237 MOVL TPASL_NUMBER(AP),R6 ; R6 = PFN TO DISPLAY
00000040'EF 56 1C AC D0 02A1 238 CMPL R6,SDA$GL_MAXPFN ; CHECK IF PFN VALID
047C'CF 00 17 1A 02A8 239 BGTRU 10$ ; BRANCH IF INVALID PFN
04CF'CF 00 00 FB 02AA 240 CALLS #0,W^PFN_TITLE ; DISPLAY THE TITLE LINE
02B4 241 CALLS #0,W^SHOW_PFN ; DISPLAY THE PFN DATA
50 01 D0 02BD 242 SKIP 1
02C0 243 MOVL #1,R0
02C1 244 RET
00000040'EF DD 02C1 245 10$: PUSHL SDA$GL_MAXPFN
02C7 246 PRINT 1,<Invalid PFN number (maximum is !XL)>
04 02D4 247 RET
02D5 248 20$: MOVL SCH$GL_FREECNT,R2 ; ADDRESS OF COUNT ARRAY
52 00000000'EF D0 02D5 250 MOVL PFNSAL_LOLIMIT,R3 ; ADDRESS OF LOLIMIT ARRAY
53 00000000'EF D0 02DC 251 MOVL PFNSAL_HILIMIT,R4 ; ADDRESS OF HILIMIT ARRAY
54 00000000'EF D0 02E3 252 MOVL PFNSAL_HEAD,R5 ; ADDRESS OF LIST HEADS
55 00000000'EF D0 02EA 253
02F1 254
19 00000000'EF 00 E1 02F1 255 BBC #OPT$V_FREE,OPTIONS,30$ ; BRANCH IF NO FREE LIST
02F9 256 SUBHD <Free page list>
0306 257 SKIP PAGE
03A3'CF 00 FB 030D 258 CALLS #0,W^SHOW_PFN_LIST ; DISPLAY FREE PAGE LIST
0312 259 30$: TSTL (R2)+
82 D5 0312 260 TSTL (R3)+
83 D5 0314 261 TSTL (R4)+
84 D5 0316 262 TSTL (R5)+
85 D5 0318 263
1B 00000000'EF 01 E1 031A 264 BBC #OPT$V_MODIFIED,OPTIONS,40$ ; BRANCH IF NO MODIFIED
0322 265 SUBHD <Modified page list>
032F 266 SKIP PAGE
A3'AF 00 FB 0336 267 CALLS #0,B^SHOW_PFN_LIST ; DISPLAY MODIFIED PAGE LIST
```

```
      82      D5      033A      268      40$:
      83      D5      033A      269
      84      D5      033C      270
      85      D5      033E      271
      18 00000000'EF 02      E1      0340      272
      02      E1      0342      273
      034A      274
      A3'AF      00      FB      0357      275
      035E      276
      0362      277
      0362      278
      0362      279
      0362      280      50$:
      31 00000000'EF 03      E1      0362      281
      00000000'EF 047C'CF 9E      036A      282
      0377      283
      0380      284
      000004CF'EF      56      D4      0387      285
      00      FB      0389      286      60$:
      56      D6      0390      287
      00000040'EF      56      D1      0392      288
      EE      1B      0399      289
      039B      290      70$:
      039B      291
      04      03A2      292
      03A3      293
      03A3      294

      TSTL      (R2)+
      TSTL      (R3)+
      TSTL      (R4)+
      TSTL      (R5)+
      BBC      #OPT$V_BAD_OPTIONS,50$ ; BRANCH IF NO BAD LIST
      SUBHD      <Bad page list>
      SKIP      PAGE
      CALLS      #0,B^SHOW_PFN_LIST ; DISPLAY BAD PAGE LIST

      PRINT ENTIRE PFN DATA FROM ENTRY 0 TO N

      BBC      #OPT$V_WHOLEPFN,OPTIONS,70$ ; BRANCH IF NOT WANTED
      SUBHD      <PFN data base>
      MOVAB      W^PFN_TITLE,HEADING_ROUTINE ; SET HEADING ROUTINE
      SKIP      PAGE
      CLRL      R6
      CALLS      #0,SHOW_PFN ; START AT PFN 0
      INCL      R6 ; SHOW PFN IN R6
      CMPL      R6,SDA$GL_MAXPFN ; SKIP TO NEXT PFN
      BLEQU      60$ ; CHECK IF LAST PFN
      ; LOOP UNTIL DONE

      STATUS      SUCCESS
      RET

      .DSABL      LSB
```



```
03A3 297 .SBTTL SHOW_PFN_LIST, DISPLAY PFN LIST
03A3 298
03A3 299
03A3 300 SHOW_PFN_LIST
03A3 301
03A3 302 THIS ROUTINE DISPLAYS THE PFN DATA FOR THE FREE,
03A3 303 MODIFIED AND BAD PAGE LISTS.
03A3 304
03A3 305 INPUTS:
03A3 306
03A3 307 R2 = ADDRESS OF COUNT LONGWORD
03A3 308 R3 = ADDRESS OF LOLIMIT LONGWORD
03A3 309 R4 = ADDRESS OF HILIMIT LONGWORD
03A3 310 R5 = ADDRESS OF LIST HEAD LONGWORD
03A3 311
03A3 312
03A3 313
03A3 314 .ENABL LSB
03A3 315
03A3 316 SHOW_PFN_LIST:
0040 03A3 317 .WORD ^M<R6>
03A3 318
03A3 319 SKIP 1
03AE 320 GETMEM (R2),-(SP) ; GET LIST COUNT
OD 50 E9 03BA 321 BLBC R0,10$
03BD 322 PRINT 1,<Count: !12SL>
03CA 323 10$:
03CA 324 GETMEM (R3),-(SP) ; GET LIST LOLIMIT
OD 50 E9 03D6 325 BLBC R0,20$
03D9 326 PRINT 1,<Lolimit: !12SL>
03E6 327 20$:
03E6 328 GETMEM (R4),-(SP) ; GET LIST HILIMIT
OD 50 E9 03F2 329 BLBC R0,30$
03F5 330 PRINT 1,<High limit: !12SL>
0402 331 30$:
0402 332 CALLS #0,B^PFN TITLE ; PRINT HEADING LINE
00000000'EF 7C'AF 00 FB 0402 333 MOVAB PFN_TITLE,HEADING_ROUTINE ; SET HEADING ROUTINE
0000047C'EF 9E 0406 334 GETMEM (R5),R6 ; GET LIST HEAD
03 50 E8 0410 335 BLBS R0,35$
0052 31 0420 336 80$: BRW 90$
0423 337 35$:
10 12 0423 338 BNEQ 40$ ; BRANCH IF NON-EMPTY LIST
0425 339 PRINT 0,<*** List is empty ***>
0040 31 0432 340 BRW 90$
0435 341 40$:
04CF'CF 00 FB 0435 342 CALLS #0,W^SHOW_PFN ; DISPLAY PFN IN R6
043A 343 PFN_REFERENCE -
043A 344 MOVAB <@SDA$X FLINK[R6],R1>,-
043A 345 LONG_OPCODE=MOVAL,-
043A 346 IMAGE=SDA
0454 347 GETMEM (R1)
15 50 E9 045D 348 BLBC R0,90$ ; SKIP IF ERROR
0460 349 PFN_REFERENCE -
0460 350 MOVZWL <R1,R6>,- ; SKIP TO NEXT ENTRY IN LIST
0460 351 LONG_OPCODE=MOVL,-
0460 352 IMAGE=SDA
03 13 0470 353 BEQL 90$ ; LOOP UNTIL END OF LIST
```

MMG
V04-000

PAGE TABLE FORMATTING ROUTINES
SHOW_PFN_LIST, DISPLAY PFN LIST

J 14

16-SEP-1984 01:35:09 VAX/VMS Macro V04-00
5-SEP-1984 03:33:12 [SDA.SRC]MMG.MAR;1

Page 11
(7)

| | | | | | | |
|-------------|----|------|-----|--------|-----------------|---------------------------------|
| FFCO | 31 | 0472 | 354 | BRW | 40\$ | |
| | | 0475 | 355 | | | |
| 00000000'EF | D4 | 0475 | 356 | CLRL | HEADING_ROUTINE | : CLEAR HEADING ROUTINE ADDRESS |
| | 04 | 047B | 357 | RET | | |
| | | 047C | 358 | | | |
| | | 047C | 359 | .DSABL | LSB | |

```
0000 047C 361 .SBTTL PFN_TITLE, DISPLAY PFN HEADING LINE
      047C 362
      047C 363
      047C 364 PFN_TITLE
      047C 365
      047C 366 DISPLAY THE HEADING LINE FOR THE PFN DATA DISPLAY
      047C 367
      047C 368
      047C 369
      047C 370 .ENABLE LOCAL_BLOCK
      047C 371
      047C 372 PFN_TITLE:
      047C 373 .WORD 0
      047E 374
      047E 375 SKIP 1
      0487 376 PFN_DISP_IF_BIGPFN_THEN
      048F
      048F 377 ;This code executes if the PFN link arrays are longword arrays.
      049C 378 PRINT 0,< PFN PTE ADDRESS BAK REFCNT FLINK BL
      04A9 379 PRINT 0,< ----
      04AB PFN_DISP_ELSE
      04AB
      04AB 380 ;This code executes if the PFN link arrays are word arrays.
      04B8 381 PRINT 0,<PFN PTE ADDRESS BAK REFCNT FLINK BLINK TY
      04C5 382 PRINT 0,<----
      04C5 PFN_DISP_ENDIF
      04C5
      04C5 ;End of code that depends on size of PFN link arrays
      04CE 383 SKIP 1
      04CF 384 RET
      04CF 385
      04CF 386 .DISABLE LOCAL_BLOCK
```

```
04CF 388 .SBTTL SHOW_PFN, SHOW DATA ON A SINGLE PFN ENTRY
04CF 389 :---
04CF 390
04CF 391 SHOW_PFN
04CF 392
04CF 393 THIS ROUTINE DISPLAYS THE PFN DATA BASE ASSOCIATED
04CF 394 WITH A SINGLE PAGE FRAME NUMBER.
04CF 395
04CF 396 INPUTS:
04CF 397
04CF 398 R6 = PAGE FRAME NUMBER
04CF 399
04CF 400 OUTPUTS:
04CF 401
04CF 402 THE ENTRY IS DISPLAYED.
04CF 403 :---
04CF 404 .ENABL LSB
04CF 405
04CF 406 SHOW_PFN:
04CF 407 .WORD 0
04CF 408
04D1 409
51 00000044'FF46 9E 04D1 410 MOVAB @SDASAB_STATE[R6],R1 ; GET PFN STATE
04D9 411 GETMEM (R1)
50 51 03 00 E9 04E2 412 BLBC R0,70$ ; SKIP IF ERROR
FC64 CF40 7F 04E5 413 EXTZV #PFNSV_LOC,#PFNSS_LOC,R1,R0
07 DD 04EA 414 PUSHAQ LOC_TABLE[R0] ; ADDRESS OF STRING
7E 51 9A 04EF 415 PUSHL #7 ; LENGTH OF STRING
51 00000048'FF46 9E 04F1 416 MOVZBL R1,-(SP)
04FC 417 MOVAB @SDASAB_TYPE[R6],R1 ; GET PFN TYPE
0505 418 GETMEM (R1)
50 51 03 00 E9 0505 419 BLBC R0,70$ ; SKIP IF ERROR
FC81 CF40 7F 0508 420 EXTZV #PFNSV_PAGTYP,#PFNSS_PAGTYP,R1,R0
07 DD 050D 421 PUSHAQ PAGTYP_TABLE[R0] ; ADDRESS OF STRING
7E 51 9A 0512 422 PUSHL #7 ; LENGTH OF STRING
0517 423 MOVZBL R1,-(SP)
0517 424 PFN REFERENCE -
0517 425 MOVAV <@SDASAx BLINK[R6],R1>,-
0517 426 LONG_OPCODE=MOVAL,-
0517 427 IMAGE=SDA
0531 428 GETMEM (R1)
33 50 E9 053A 429 70$: BLBC R0,80$ ; SKIP IF ERROR
053D 430 PFN REFERENCE -
053D 431 MOVZWL <R1,-(SP)>,- ; BACKWARD LINK
053D 432 LONG_OPCODE=MOVL,-
053D 433 IMAGE=SDA
054D 434 PFN REFERENCE -
054D 435 MOVAV <@SDASAx FLINK[R6],R1>,-
054D 436 LONG_OPCODE=MOVAL,-
054D 437 IMAGE=SDA
0567 438 GETMEM (R1)
7B 50 E9 0570 439 80$: BLBC R0,90$ ; SKIP IF ERROR
0573 440 PFN REFERENCE -
0573 441 MOVZWL <R1,-(SP)>,- ; FORWARD LINK
0573 442 LONG_OPCODE=MOVL,-
0573 443 IMAGE=SDA
51 0000004C'FF46 3E 0583 444 MOVAV @SDASAW_REFCNT[R6],R1
```



```

      57 50  E9 059B 445  GETMEM (R1)
      7E 51  3C 0594 446  BLBC   R0,90$           ; SKIP IF ERROR
51 00000050'FF46 DE 0597 447  MOVZWL R1,-(SP)       ; REFERENCE COUNT
      3D 50  E9 059A 448  MOVAL   @SDASAL BAK[R6],R1
      56  DD  C5A2 449  GETMEM (R1),-(SP)       ; BACKING STORE ADDRESS
51 00000054'FF46 DE 05AE 450  BLBC   R0,90$           ; SKIP IF ERROR
      26 50  E9 05B1 451  MOVAL   @SDASAL PTE[R6],R1
      56  DD  05B9 452  GETMEM (R1),-(SP)       ; ADDRESS OF PAGE TABLE ENTRY
      56  DD  05C3 453  BLBC   R0,90$           ; SKIP IF ERROR
      56  DD  05C8 454  PUSHL   R6              ; PFN INDEX
      56  DD  05CA 455  PFN_DISP_IF_BIGPFN_THEN ; If greater than 32 Mbytes, then use longwo
      56  DD  05D2
      56  DD  05D2 456  ;This code executes if the PFN Link arrays are longword arrays.
      56  DD  05DF 457  PRINT 12,<!XL !XL !XL !5UW !XL !XL !XB !AD !XB !
      56  DD  05E1
      56  DD  05E1
      56  DD  05E1 458  ;This code executes if the PFN Link arrays are word arrays.
      56  DD  05EE 459  PRINT 12,<!XW !XL !XL !5UW !XW !XW !XB !AD !XB !
      56  DD  05EE
      56  DD  05EE
      04 05EE 460 90$: ;End of code that depends on size of PFN Link arrays
      56  DD  05EF 461  RET
      56  DD  05EF 462  .DSABL LSB
```

```
05EF 465 .SBTTL DISPLAY_SPT_RANGE -- DISPLAY SYSTEM PAGE TABLE W/RANGE
05EF 466 :---
05EF 467 :
05EF 468 DISPLAY_SPT_RANGE
05EF 469 :
05EF 470 THIS ROUTINE FORMATS THE ENTIRE CONTENTS OF THE SYSTEM
05EF 471 PAGE TABLE, OR ANY SUBRANGE THEREOF.
05EF 472 :
05EF 473 INPUTS:
05EF 474 :
05EF 475 OPTIONS = OPTIONS FLAGS (RANGE OR LENGTH BITS RELEVANT)
05EF 476 ESP = START OF PAGE TABLE VA
05EF 477 (OR, IF LENGTH BIT SET)
05EF 478 ESP = SIZE OF PAGE TABLE VA
05EF 479 ESP+4 = HIGH LIMIT OF PAGE TABLE VA
05EF 480 :
05EF 481 OUTPUTS:
05EF 482 :
05EF 483 NONE
05EF 484 :
05EF 485 :---
05EF 486 :
003C 05EF 487 .ENTRY DISPLAY_SPT_RANGE,^M<R2,R3,R4,R5>
05F1 488 :
50 00000000'EF 9E 05F1 489 MOVAB OPTIONS, R0 ; POINT TO OPTIONS WORD
52 60 D0 05F8 490 (R0), R2
51 00000000'EF D0 05FB 491 3$: MOVL ESP, R1 ; POINT TO EXPRESSION STACK
07 52 03 E0 0602 492 BBS #OPT$V_RANGE, R2, 10$ ; RANGE SPECIFIED
11 52 04 E0 0606 493 BBS #OPT$V_LENGTH, R2, 20$ ; LENGTH SPECIFIED
50 D4 060A 494 5$: CLRL R0 ; SYNTAX ERROR
04 060C 495 RET
060D 496 :
54 04 A1 D0 060D 497 10$: MOVL 4(R1), R4 ; R4 = LOWEST ADDRESS
53 61 54 C3 0611 498 SUBL3 R4, (R1), R3 ; R3 = SIZE
05 60 04 E2 0615 499 BBSS #OPT$V_LENGTH, (R0), 30$ ; SET A SINGLE BIT FOR RANGE
03 11 0619 500 BRB 30$
061B 501 :
53 61 7D 061B 502 20$: MOVQ (R1), R3 ; R4 = LOWEST ADDRESS
061E 503 :
54 01FF 8F AA 061E 504 30$: BICW #^X1FF, R4 ; ROUND DOWN
53 000001FF 8F C0 0623 505 ADDL2 #^X1FF, R3
53 53 F7 8F 78 062A 506 ASHL #-9, R3, R3 ; MAKE NUMBER OF ENTRIES
0A 11 062F 507 BRB DISP ; JOIN COMMON CODE
```

```
0631 509 .SBTTL DISPLAY_SPT DISPLAY SYSTEM PAGE TABLE
0631 510 ---
0631 511
0631 512 DISPLAY_SPT
0631 513
0631 514 THIS ROUTINE FORMATS THE ENTIRE CONTENTS OF THE SYSTEM
0631 515 PAGE TABLE.
0631 516
0631 517 INPUTS:
0631 518 NONE
0631 519
0631 520 OUTPUTS:
0631 521 NONE
0631 522
0631 523 ---
0631 524
0631 525 .ENABL LSB
0631 526
0631 527 003C .ENTRY DISPLAY_SPT,^M<R2,R3,R4,R5>
0631 528
0631 529 BBCC #OPT$V_LENGTH,OPTIONS,DISP ; CLEAR IT, IF SET BY /ALL
0631 530 CALLS #0,INIT_PFN ; SETUP TO READ PFN DATA
0631 531 DISP:
0631 532
0631 533 DISPLAY THE SYSTEM PAGE TABLE
0631 534
0631 535 BBC #OPT$V_SYSTEM,OPTIONS,10$ ; BRANCH IF NOT SELECTED
0631 536 SUBHD <System page table>
0631 537 SKIP PAGE
0631 538 GETMEM @MMG$GL_SYSPHD ; ADDRESS OF SYSPHD
0631 539 BLBS R0,5$ ; Branch if ok...else
0631 540 BRW 90$ ;...Return
0631 541 5$:
0631 542 PUSHL R1
0631 543 MOVAB BUFFER,R2
0631 544 GETMEM PHD$$_POBR(R1),(R2),#8 ; GET VIRTUAL SBR,SLR
0631 545 BLBS R0,6$ ; OKAY
0631 546 BRW 90$ ; BRANCH IF ERROR
0631 547 6$:
0631 548 PUSHL (R2) ; STARTING ADDRESS
0631 549 MOVL #^X80000000,R5 ; STARTING ADDRESS BEING MAPPED
0631 550 BBS #OPT$V_LENGTH,OPTIONS,7$ ; IF RANGE NOT SPECIFIED...
0631 551 EXTZV #PHD$$_POLR,#PHD$$_POLR,4(R2),R3 ; #ENTRIES
0631 552 MOVL R5,R4 ; STARTING ADDRESS
0631 553 7$:
0631 554 SUBL3 R5,R4,R5 ; OFFSET INTO AREA
0631 555 ASHL #-7,R5,R5 ; TURN INTO NUMBER OF ENTRIES TO SKIP
0631 556 ADDL R5,(SP) ; UPDATE START ENTRY
0631 557 MOVQ R3,-(SP) ; #ENTRIES,START ADDR
0631 558 CALLS #4,B^DUMP_PTE ; FORMAT PAGE TABLE
0631 559 BBCC #OPT$V_LENGTH,OPTIONS,10$ ; CLEAR IT OUT
0631 560
0631 561 DISPLAY THE GLOBAL PAGE TABLE
0631 562
0631 563 10$:
0631 564 BBC #OPT$V_GLOBAL,OPTIONS,90$ ; BRANCH IF NOT SELECTED
0631 565 SUBHD <Global page table>
0631 566 SKIP PAGE
0631 567 GETMEM @MMG$GL_SYSPHD,-(SP) ; ADDRESS OF PROCESS HEADER
0631 568 GETMEM @MMG$GL_GPTE,R2 ; ADDRESS OF FIRST GPTE
0631 569 BLBS R0,90$
```

00 00000000'EF 04 E5
FB97 CF 00 FB

7A 00000000'EF 02 E1

03 50 E8
00C1 31
51 DD
52 00000000'EF 9E

03 50 E8
00A3 31
62 DD
55 80000000 8F DO
09 00000000'EF 04 E0
53 04 A2 18 00 EF
54 55 DO
55 54 55 C3
55 55 F9 8F 78
6E 55 C0
7E 53 7D
38'AF 04 FB
00 00000000'EF 04 E5

66 00000000'EF 00 E1

2F 50 E9

| | | | | | | | | | | | |
|----|----------|----|----|----|----|------|-----|--------|-----------------------------|---|-------------------------------------|
| 06 | 00000000 | EF | 1F | 50 | E9 | 0701 | 566 | GETMEM | @MMG\$GL_MAXGPT | : | ADDRESS OF LAST+1 GPT |
| | 53 | 51 | | 52 | DD | 070E | 567 | BLBC | R0,90\$ | | |
| | | | | 04 | E0 | 0711 | 568 | PUSHL | R2 | : | STARTING ADDRESS OF PAGE TABLE |
| | | | | 52 | C3 | 0713 | 569 | BBS | #OPT\$V,LENGTH,OPTIONS,30\$ | : | IF RANGE NOT SPECIFIED... |
| | | | | 54 | D4 | 071B | 570 | SUBL3 | R2,R1,R3 | : | LENGTH OF PAGE TABLE |
| | | | | 8F | 78 | 071F | 571 | CLRL | R4 | : | FIRST PAGETABLE ENTRY |
| 55 | 54 | F9 | | 55 | CO | 0721 | 572 | ASHL | #-7,R4,R5 | : | TURN INTO NUMBER OF ENTRIES TO SKIP |
| | | | | 53 | 7D | 0726 | 573 | ADDL | R5,(SP) | : | UPDATE START ENTRY |
| | | | | 04 | FB | 0729 | 574 | MOVQ | R3,-(SP) | : | |
| | | | | | | 072C | 575 | CALLS | #4,B^DUMP_PTE | : | FORMAT PAGE TABLE |
| | | | | | | 0730 | 576 | | | | |
| | | | | | | 0730 | 577 | STATUS | SUCCESS | | |
| | | | | | | 0737 | 578 | RET | | | |
| | | | | | | 0738 | 579 | | | | |
| | | | | | | 0738 | 580 | .DSABL | LSB | | |


```
0738 583 .SBTTL DUMP_PTE -- FORMAT THE PAGE TABLE
0738 584
0738 585
0738 586
0738 587
0738 588
0738 589
0738 590
0738 591
0738 592
0738 593
0738 594
0738 595
0738 596
0738 597
0738 598
0738 599
0738 600
0738 601
0738 602
0738 603
0738 604
0738 605
0738 606
0738 607
0738 608
0738 609
0738 610
073A 611
073A 612
073A 613
073A 614
073F 615
0748 616
074A 617
074D 618
074F 619
0750 620
0750 621
0754 622
0757 623
0757 624
0757 625
0757 626
0758 627
075F 628
0769 629
076C 630
076F 631
0772 632
0775 633
0775 634
0778 635
077A 636
077D 637
0780 638
0783 639

00000060
07FC

0A0B'CF 00 FB
00000000'EF 0A0B'CF 9E
          59 7C
          04 AC D5
          01 14
          04 074F
          0750
5E A0 AE 9E
52 SE D0

82 08 AC D0
82 0C AC D0
          09 50 E8
50 01 D0
          02D2 30
          0271 31

53 51 D0
          09 12
50 02 D0
          02C4 30
          0263 31

          0748
          074A
          074D
          074F
          0750
          0754
          0757
          0757
          0757
          075F
          0769
          076C
          076F
          0772
          0775
          0775
          0778
          077A
          077D
          0780
          0783

          10$:
          20$:
          30$:

          SCRATCH_SIZE = 24*4 ; 24 LONGWORDS
          .ENTRY DUMP_PTE, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10>
          .ENABL LSB
          CALLS #0, W^PTE_TITLE ; PRINT SUB-HEADING LINE
          MOVAB W^PTE_TITLE, HEADING_ROUTINE ; SET HEADING ROUTINE
          CLRQ R9 ; INITIALIZE STATE TO NORMAL
          TSTL 4(AP) ; CHECK IF ANY TO DUMP
          BGTR 10$ ; BRANCH IF SO
          RET
          MOVAB -SCRATCH_SIZE(SP), SP ; RESERVE SPACE FOR FAO PARAMS
          MOVL SP, R2 ; R2 USED TO STORE PARAMS
          FORMAT THE PAGE TABLE ENTRY
          MOVL 8(AP), (R2)+ ; MAPPING ADDRESS
          MOVL 12(AP), (R2)+ ; VIRTUAL ADDRESS OF ENTRY
          TRYMEM @12(AP) ; GET PAGE TABLE ENTRY
          BLBS R0, 20$ ; IF ENTRY FOUND
          MOVL #1, R0
          BSBW PTE_STATE ; SET STATE = 1 (INVALID MEMORY)
          BRW 80$ ; AND SKIP THIS ENTRY
          MOVL R1, R3 ; SAVE PTE IN R3
          BNEQ 30$ ; BRANCH IF NOT NULL PAGE
          MOVL #2, R0
          BSBW PTE_STATE ; SET STATE = 2 (NULL PAGES)
          BRW 80$ ; AND SKIP THIS ENTRY
```

```

      50      D4 0783      640      CLRL      R0
02BC      30 0785      641      BSBW      PTE_STATE      ; SET STATE TO NORMAL
      0788      642
58      53      15      00      EF 0788      643      EXTZV      #PTESV_PFN,#PTES_PFN,R3,R8      ; GET PFN IF PRESENT
      54      05      D0 078D      644      MOVL      #5,R4      ; TYPE CODE FOR VALID
      82      53      D0 0790      645      MOVL      R3,(R2)+      ; STORE PTE IN FAO LIST
      18      18      0793      646      BGEQ      32$      ; BRANCH IF NOT VALID
00000000'EF      58      D1 0795      647      CMPL      R8,PHYS_PAGES      ; CHECK IF LEGAL
      08      18      079C      648      BGEQ      31$      ; BRANCH IF INVALID PFN
00000040'EF      58      D1 079E      649      CMPL      R8,SDASGL_MAXPFN      ; CHECK IF WITHIN PFN DATABASE
      1D      1A      07A5      650      BGTRU      36$      ; BYPASS PFN LOOKUP IF SO
      1E      11      07A7      651      BRB      40$      ; GOOD PFN
      07A9      652      31$:
      54      D6      07A9      653      INCL      R4      ; TYPE CODE FOR I/O PAGE
      17      11      07AB      654      BRB      36$      ; AND INDICATE INVALID PFN
      07AD      655      32$:
54      53      01      16      EF 07AD      656      EXTZV      #PTESV_TYPO,#1,R3,R4      ; BRING TYPO AND TYP1
      03      53      1A      E1 07B2      657      BBC      #PTESV_TYP1,R3,34$      ; TOGETHER
      54      02      C8      07B6      658      BISL      #2,R4      ; SET HIGH ORDER BIT
      07B9      659      34$:
      54      D5      07B9      660      TSTL      R4      ; 0 = TRANSITION OR DZERO
      07      12      07BB      661      BNEQ      36$      ; BRANCH IF NOT
      58      D5      07BD      662      TSTL      R8      ; PFN SHOULD BE 0 FOR DZERO
      06      12      07BF      663      BNEQ      40$      ; BRANCH IF TRANSITION
      54      04      D0      07C1      664      MOVL      #4,R4      ; TYPE CODE FOR DZERO
      07C4      665      36$:
      58      01      CE      07C4      666      MNEGL      #1,R8      ; INDICATE NO PFN
      07C7      667      :
      07C7      668      :
      07C7      669      :
      07C7      670      :
      07C7      671      :
      07C7      672      40$:
      54      05      C4      07C7      673      MULL2      #5,R4      ; INDEX INTO TYPE TABLE
      82      05      D0      07CA      674      MOVL      #5,(R2)+      ; LENGTH OF STRING
      82      F959      CF44      9E      07CD      675      MOVAB      TYPE_TABLE[R4],(R2)+      ; ADDRESS OF STRING
      56      7C      07D3      676      CLRQ      R6      ; ASSUME MODIFY/LOCK BITS OFF
      4C      53      1F      E1      07D5      677      BBC      #PTESV_VALID,R3,45$      ; BRANCH IF NOT VALID
57      53      01      1A      EF 07D9      678      EXTZV      #PTESV_MODIFY,#1,R3,R7      ; GET MODIFY BIT FROM PTE
      43      58      1F      E0      07DE      679      BBS      #31,R8,45$      ; BRANCH IF NO PFN
      3E      08      AC      1F      E0      07E2      680      BBS      #31,8(AP),45$      ; BRANCH IF SPT
      07E7      681      PFN REFERENCE -
      07E7      682      MOVAB      <@SDASGL_WSLX[R8],R1>,-      ; ADDRESS OF WSLX FIELD
      07E7      683      LONG_OPCODE=MOVAL,-
      07E7      684      IMAGE=SDA
      0801      685      GETMEM      (R1)      ; GET LONGWORD
      18      50      E9      080A      686      BLBC      R0,45$      ; IF NOT FOUND
      51      51      32      080D      687      CVTWL      R1,R1      ; EXTEND FIELD
      13      13      0810      688      BEQL      45$      ; BRANCH IF NOT A WSL INDEX
      51      10      BC41      DE      0812      689      MOVAL      @16(AP)[R1],R1      ; ADDRESS OF WSL ENTRY
      0817      690      GETMEM      (R1)      ; GET WSL LONGWORD
56      51      01      05      EF 0820      691      EXTZV      #WSLSV_WSLock,#1,R1,R6      ; WSL LOCK BIT
      0825      692      45$:
51      53      04      18      EF 0825      693      EXTZV      #PTESV_PROT,#PTES_PROT,R3,R1      ; GET PROTECTION CODE
      82      04      D0      082A      694      MOVL      #4,(R2)+      ; LENGTH OF STRING
      82      F885      CF41      DE      082D      695      MOVAL      PROT_TABLE[R1],(R2)+      ; PAGE PROTECTION
      82      01      D0      0833      696      MOVL      #1,(R2)+      ; SIZE OF MODIFY STRING
```

```
      82 F998 CF47 9E 0836 697      MOVAB MODIFY_TABLE[R7],(R2)+ ; ADDRESS OF STRING
      82 82 01 DO 083C 698      MOVL #1,(R2)+ ; SIZE OF WSLOCK STRING
51 82 F991 CF46 9E 083F 699      MOVAB WSLOCK_TABLE[R6],(R2)+ ; ADDRESS OF STRING
      53 02 17 EF 0845 700      EXTZV #PTESV_OWN,#PTES$ _OWN,R3,R1 ; GET PAGE OWNER
      82 82 01 DO 084A 701      MOVL #1,(R2)+ ; LENGTH OF STRING
      82 F8D5 CF41 9E 084D 702      MOVAB OWNER_TABLE[R1],(R2)+ ; ADDRESS OF STRING
      24 58 1F E1 0853 703      BBC #31,R8,50$ ; BRANCH IF PFN VALID
00000000'EF 00 FB 0871 705      $FAOL_S PTECTL1,LIST+RAB$W_RSZ,LINE_DESCR,-SCRATCH_SIZE(FP)
      016B 31 0878 706      CALLS #0,PUT_LINE ; OUTPUT LINE
      087B 707      BRW 80$ ; SKIP TO NEXT ENTRY
      087B 708      :
      087B 709      :
      087B 710 50$:
51 00000048'FF48 9E 087B 711      MOVAB @SDASAB_TYPE[R8],R1
      58 50 E9 0883 712      GETMEM (R1)
      51 51 03 00 EF 088C 713      BLBC R0,70$ ; SKIP IF ERROR
      82 07 DO 088F 714      EXTZV #PFNSV_PAGTYP,#PFNS$ _PAGTYP,R1,R1 ; GET PAGE TYPE
      82 F8F7 CF41 7E 0894 715      MOVL #7,(R2)+ ; LENGTH OF STRING
51 00000044'FF48 9E 089D 716      MOVAQ PAGTYP_TABLE[R1],(R2)+ ; ADDRESS OF STRING
      08A5 717      MOVAB @SDASAB_STATE[R8],R1
      39 50 E9 08AE 718      GETMEM (R1)
      51 51 03 00 EF 08B1 719      BLBC R0,70$ ; SKIP IF ERROR
      82 07 DO 08B6 720      EXTZV #PFNSV_LOC,#PFNS$ _LOC,R1,R1 ; GET PAGE LOCATION
      82 F895 CF41 7E 08B9 721      MOVL #7,(R2)+ ; LENGTH OF STRING
51 00000044'FF48 9E 08BF 722      MOVAQ LOC_TABLE[R1],(R2)+ ; ADDRESS OF STRING
      08C7 723      MOVAB @SDASAB_STATE[R8],R1
      17 50 E9 08D0 724      GETMEM (R1) ; GET STATE FIELD
      82 51 9A 08D3 725      BLBC R0,70$ ; SKIP IF ERROR
51 00000048'FF48 9E 08D6 726      MOVZBL R1,(R2)+
      03 50 EB 08E7 727      MOVAB @SDASAB_TYPE[R8],R1
      00F9 31 08EA 728      GETMEM (R1) ; GET TYPE FIELD
      82 51 9A 08ED 729      BLBS R0,71$
51 0000004C'FF48 3E 08F0 730 70$:
      E6 50 E9 0901 731 71$:
      82 51 3C 0904 732      MOVZBL R1,(R2)+
51 00000050'FF48 DE 0907 733      MOVAW @SDASAW_REFCNT[R8],R1
      CF 50 E9 0918 734      GETMEM (R1) ; COUNT OF PAGE REFERENCES
      82 51 DO 091B 735      BLBC R0,70$ ; SKIP IF ERROR
      82 51 DE 091E 736      MOVZWL R1,(R2)+
      090F 737      MOVAL @SDASAL_BAK[R8],R1
      0918 738      GETMEM (R1) ; BACKING STORE ADDRESS
      82 51 DO 091B 739      BLBC R0,70$ ; SKIP IF ERROR
51 00000054'FF48 DE 091E 740      MOVAL @SDASAL_PTE[R8],R1
      B8 50 E9 092F 741      GETMEM (R1) ; ADDRESS OF PTE
      82 51 DO 0932 742      BLBC R0,70$ ; SKIP IF ERROR
      0935 743      MOVL R1,(R2)+
      0935 744      PFN REFERENCE -
      0935 745      MOVAV <@SDASAx FLINK[R8],R1>,-
      0935 746      LONG_OPCODE=MOVAL,-
      0935 747      IMAGE=SDA
      8F 50 E9 0958 748      GETMEM (R1) ; FORWARD PAGE LIST LINK
      0958 749      BLBC R0,70$ ; SKIP IF ERROR
      0958 750      PFN REFERENCE -
      0958 751      MOVZWL <R1,(R2)>,-
      0958 752      LONG_OPCODE=MOVL,-
      0958 753      IMAGE=SDA
```

```
096B 754
096B 755
096B 756
096B 757
0985 758
55 50 E9 098E 759
0991 760
0991 761
0991 762
0991 763
09A1 764
09A1 765
09A9
09A9
09A9 766
09C3 767
09C3 768
09C5
09C5
09C5 769
09DF 770
09DF
09DF
00000000'EF 00 FB 09DF 771
09E6 772
09E6 773
09E6 774
09E6 775
09E6 776
SE 60 AE 9E 09EA 777
OC AC 04 C0 09EE 778
OB AC 00000200 8F C0 09F6 779
04 AC D7 09F9 780
03 15 09FB 781
FD52 31 09FE 782
50 D4 09FE 783
0041 30 0A00 784
0A03 785
0A0A 786
0A0B 787
0A0B 788
0A0B 789
0A0B 790
0A0B 791
0A0B 792
0A0B 793
0A0B 794
0A0B 795
0000 0A0B 796
0A0B 797
0A0D 798
0A0D 799
0A16 800
0A1E
0A1E
0A1E 801
0A1E 802

PFN_REFERENCE -
MOVAV <@SDA$ax BLINK[R8],R1>,-
LONG_OPCODE=MOVAL,-
IMAGE=SDA
GETMEM (R1) ; BACKWARD PAGE LIST LINK
BLBC R0,80$ ; SKIP IF ERROR
MOVZWL <R1-(R2)+>,-
LONG_OPCODE=MOVL,-
IMAGE=SDA
PFN_DISP_IF_BIGPFN_THEN ; For larger than 32 Mbytes, use longword fo
END_BIGPFN_CODE=74$
;This code executes if the PFN link arrays are longword arrays.
$FAOL_S PTECTL2_LONG,LIST+RABSW_RSZ,LINE_DESCR,-SCRATCH_SIZE(FP)
; Otherwise, use word format
PFN_DISP_ELSE ELSE_CODE=74$ , COMMON_CODE=77$
;This code executes if the PFN link arrays are word arrays.
$FAOL_S PTECTL2_WORD,LIST+RABSW_RSZ,LINE_DESCR,-SCRATCH_SIZE(FP)
PFN_DISP_ENDIF COMMON_CODE=77$
;End of code that depends on size of PFN link arrays
CALLS #0,PUT_LINE ; OUTPUT LINE
SKIP TO NEXT PAGE TABLE ENTRY
80$:
MOVAB SCRATCH_SIZE(SP),SP ; DEALLOCATE FAO SPACE
ADDL2 #4,12(AP) ; NEXT PTE
ADDL2 #512,8(AP) ; INCREMENT MAPPING ADDRESS
DECL 4(AP) ; DECREMENT REPEAT COUNT
BLEQ 90$ ; EXIT IF DONE
BRW 10$
90$:
CLRL R0
BSBW PTE_STATE ; TERMINATE CURRENT STATE
STATUS SUCCESS
RET
.DSABL LSB
SUBROUTINE TO PRINT THE SUB-HEADING LINE
.ENABLE LOCAL_BLOCK
PTE_TITLE:
.WORD 0
SKIP 1
PFN_DISP_IF_BIGPFN_THEN ; For larger than 32 Mbytes, use longword fo
;This code executes if the PFN link arrays are longword arrays.
PRINT 0,-
<!,_ ADDRESS SVAPTE PTE TYPE PROT BITS PAGTYP LOC S
```



```
0A28 803 PFN_DISP_ELSE ; Otherwise, use word format
0A2D
0A2D ;This code executes if the PFN link arrays are word arrays.
0A2D 804 PRINT 0, -
0A2D 805 <! ADDRESS SVAPTE PTE TYPE PROT BITS PAGTYP LOC S
0A3A 806 PFN_DISP_ENDIF
0A3A
0A3A ;End of code that depends on size of PFN link arrays
04 0A3A 807 SKIP 1
0A43 808 RET
0A44 809
0A44 810 .DISABLE LOCAL_BLOCK
```

```
0A44 813 .SBTTL PTE_STATE SET STATE OF PTE DISPLAY
0A44 814 ---
0A44 815
0A44 816 PTE_STATE
0A44 817
0A44 818 SET STATE OF RUNNING SCAN OF PAGE TABLE AND PRINT ANY
0A44 819 STATUS MESSAGES FROM THE PREVIOUS STATE.
0A44 820
0A44 821 INPUTS:
0A44 822
0A44 823 R0 = REQUESTED NEW STATE
0A44 824 R9 = CURRENT STATE
0A44 825 R10 = REPITITION COUNT IN SAME STATE
0A44 826
0A44 827 OUTPUTS:
0A44 828
0A44 829 R9 = NEW STATE
0A44 830 R10 = UPDATED REPITITION COUNT
0A44 831
0A44 832 ---
0A44 833
0A44 834 .ENABL LSB
0A44 835
0A44 836 PTE_STATE:
59 50 D1 0A44 837 CMPL R0,R9 ; CHECK IF ALREADY IN STATE
03 12 0A47 838 BNEQ 10$ ; BRANCH IF NOT
5A D6 0A49 839 INCL R10 ; INCREMENT REPITITION COUNT
05 05 0A4B 840 RSB
0A4C 841 10$:
01 50 DD 0A4C 842 PUSHL R0 ; SAVE NEW STATE
59 D1 0A4E 843 CMPL R9,#1 ; CHECK IF BYPASSING BAD MEMORY
23 12 0A51 844 BNEQ 20$ ; BRANCH IF NOT
0A53 845 SKIP 1
5A DD 0A5C 846 PUSHL R10
0A5E 847 PRINT 1,<!-- !UL ENTRIES NOT IN MEMORY>
0A6B 848 SKIP 1
26 11 0A74 849 BRB 80$
0A76 850 20$:
02 59 D1 0A76 851 CMPL R9,#2 ; CHECK IF SKIPPING NULL PAGES
21 12 0A79 852 BNEQ 80$ ; BRANCH IF NOT
0A7B 853 SKIP 1
5A DD 0A84 854 PUSHL R10
0A86 855 PRINT 1,<!-- !UL NULL PAGE!XS>
0A93 856 SKIP 1
0A9C 857 80$:
SA 59 BEDD 0A9C 858 POPL R9 ; SET NEW STATE
01 D0 0A9F 859 MOVL #1,R10 ; INITIALIZE REPITITION COUNTER
05 0AA2 860 RSB
0AA3 861
0AA3 862 .DSABL LSB
```

MMG
V04-000

PAGE TABLE FORMATTING ROUTINES J 15
PTE_STATE SET STATE OF PTE DISPLAY

16-SEP-1984 01:35:09 VAX/VMS Macro V04-00
5-SEP-1984 03:33:12 [SDA.SRC]MMG.MAR;1

Page 24
(14)

0AA3 864
0AA3 865 .END

MMG
Symbol table

PAGE TABLE FORMATTING ROUTINES

K 15

16-SEP-1984 01:35:09 VAX/VMS Macro V04-00
5-SEP-1984 03:33:12 [SDA.SRC]MMG.MAR;1

Page 25
(14)

| | | | |
|-------------------|------------|----|----|
| ARGS | = 00000001 | | |
| BUFFER | 00000000 | R | 02 |
| DISP | 0000063B | R | 03 |
| DISPLAY_PFN | 0000028E | RG | 03 |
| DISPLAY_SPT | 00000631 | RG | 03 |
| DISPLAY_SPT_RANGE | 000005EF | RG | 03 |
| DUMP_PTE | 00000738 | RG | 03 |
| ESP | ***** | X | 03 |
| GETMEM | ***** | X | 03 |
| HEADING_ROUTINE | ***** | X | 03 |
| INIT_PFN | 000001D7 | RG | 03 |
| LINE_DESCR | ***** | X | 03 |
| LIST | ***** | X | 03 |
| LOC_TABLE | 00000153 | R | 03 |
| MMG\$GL_GPTE | ***** | X | 03 |
| MMG\$GL_MAXGPTE | ***** | X | 03 |
| MMG\$GL_MAXPFN | ***** | X | 03 |
| MMG\$GL_SYSPHD | ***** | X | 03 |
| MMG\$GW_BIGPFN | = 00000042 | R | 02 |
| MODIFY_TABLE | 000001D3 | R | 03 |
| MSG\$ SUCCESS | ***** | X | 03 |
| NEW PAGE | ***** | X | 03 |
| OPTSV_BAD | = 00000002 | | |
| OPTSV_FREE | = 00000000 | | |
| OPTSV_GLOBAL | = 00000000 | | |
| OPTSV_LENGTH | = 00000004 | | |
| OPTSV_MODIFIED | = 00000001 | | |
| OPTSV_RANGE | = 00000003 | | |
| OPTSV_SINGLEPFN | = 00000004 | | |
| OPTSV_SYSTEM | = 00000002 | | |
| OPTSV_WHOLEPFN | = 00000003 | | |
| OPTIONS | ***** | X | 03 |
| OWNER_TABLE | 00000127 | R | 03 |
| PAGTYP_TABLE | 00000193 | R | 03 |
| PFNSAB_STATE | ***** | X | 03 |
| PFNSAB_TYPE | ***** | X | 03 |
| PFNSAL_BAK | ***** | X | 03 |
| PFNSAL_HEAD | ***** | X | 03 |
| PFNSAL_HI LIMIT | ***** | X | 03 |
| PFNSAL_LO LIMIT | ***** | X | 03 |
| PFNSAL_PTE | ***** | X | 03 |
| PFNSAW_REFCNT | ***** | X | 03 |
| PFNSAX_BLINK | ***** | X | 03 |
| PFNSAX_FLINK | ***** | X | 03 |
| PFNSAX_WSLX | ***** | X | 03 |
| PFNSS_LOC | = 00000003 | | |
| PFNSS_PAGTYP | = 00000003 | | |
| PFNSV_LOC | = 00000000 | | |
| PFNSV_PAGTYP | = 00000000 | | |
| PFN TITLE | 0000047C | R | 03 |
| PHD\$ POLR | = 000000C8 | | |
| PHD\$ POLR | = 00000018 | | |
| PHD\$ POLR | = 00000000 | | |
| PHYS PAGES | ***** | X | 03 |
| PRINT | ***** | X | 03 |
| PROT_TABLE | 000000E7 | R | 03 |
| PTE\$ OWN | = 00000002 | | |

| | | | |
|-----------------|------------|----|----|
| PTE\$ PFN | = 00000015 | | |
| PTE\$ PROT | = 00000004 | | |
| PTE\$V MODIFY | = 0000001A | | |
| PTE\$V OWN | = 00000017 | | |
| PTE\$V PFN | = 00000000 | | |
| PTE\$V PROT | = 0000001B | | |
| PTE\$V TYP0 | = 00000016 | | |
| PTE\$V TYP1 | = 0000001A | | |
| PTE\$V VALID | = 0000001F | | |
| PTECTL1 | 00000000 | R | 03 |
| PTECTL2_LONG | 0000008B | R | 03 |
| PTECTL2_WORD | 0000002F | R | 03 |
| PTE_STATE | 00000A44 | R | 03 |
| PTE_TITLE | 00000A0B | R | 03 |
| PUT_LINE | ***** | X | 03 |
| RAB\$W RSZ | ***** | X | 03 |
| REQMEM | ***** | X | 03 |
| SCH\$GL FREECNT | ***** | X | 03 |
| SCRATCH_SIZE | = 00000060 | | |
| SDASAB_STATE | 00000044 | R | 02 |
| SDASAB_TYPE | 00000048 | R | 02 |
| SDASAL_BAK | 00000050 | R | 02 |
| SDASAL_PTE | 00000054 | R | 02 |
| SDASAW_REFCNT | 0000004C | R | 02 |
| SDASAX_BLINK | 0000005C | R | 02 |
| SDASAX_FLINK | 00000058 | R | 02 |
| SDASAX_WSLX | 00000060 | R | 02 |
| SDASGL_MAXPFN | 00000040 | R | 02 |
| SET HEADING | ***** | X | 03 |
| SHOW_PFN | 000004CF | R | 03 |
| SHOW_PFN_LIST | 000003A3 | R | 03 |
| SKIP_LINES | ***** | X | 03 |
| SYSS\$FAOL | ***** | GX | 03 |
| TPASL NUMBER | = 0000001C | | |
| TRYMEM | ***** | X | 03 |
| TYPE TABLE | 0000012B | R | 03 |
| WSL\$W WSLOCK | = 00000005 | | |
| WSLOCK_TABLE | 000001D5 | R | 03 |

+-----+
! Psect synopsis !
+-----+

| PSECT name | Allocation | PSECT No. | Attributes |
|------------|-------------------|-----------|---|
| . ABS . | 00000000 (0.) | 00 (0.) | NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE |
| \$AB\$\$ | 00000000 (0.) | 01 (1.) | NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE |
| SDADATA | 00000064 (100.) | 02 (2.) | NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC BYTE |
| MMG | 00000AA3 (2723.) | 03 (3.) | NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE |
| LITERALS | 00000462 (1122.) | 04 (4.) | NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE |

+-----+
! Performance indicators !
+-----+

| Phase | Page faults | CPU Time | Elapsed Time |
|------------------------|-------------|-------------|--------------|
| Initialization | 30 | 00:00:00.03 | 00:00:01.48 |
| Command processing | 112 | 00:00:00.43 | 00:00:03.01 |
| Pass 1 | 353 | 00:00:07.85 | 00:00:26.11 |
| Symbol table sort | 0 | 00:00:00.97 | 00:00:05.81 |
| Pass 2 | 163 | 00:00:02.21 | 00:00:09.83 |
| Symbol table output | 11 | 00:00:00.06 | 00:00:00.33 |
| Psect synopsis output | 3 | 00:00:00.02 | 00:00:00.02 |
| Cross-reference output | 0 | 00:00:00.00 | 00:00:00.00 |
| Assembler run totals | 674 | 00:00:11.57 | 00:00:46.59 |

The working set limit was 1500 pages.
76303 bytes (150 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 875 non-local and 96 local symbols.
865 source lines were read in Pass 1, producing 41 object records in Pass 2.
31 pages of virtual memory were used to define 29 macros.

+-----+
! Macro library statistics !
+-----+

| Macro library name | Macros defined |
|-------------------------------------|----------------|
| _\$255\$DUA28:[SDA.OBJ]SDALIB.MLB;1 | 9 |
| _\$255\$DUA28:[SYS.OBJ]LIB.MLB;1 | 9 |
| _\$255\$DUA28:[SYSLIB]STARLET.MLB;2 | 8 |
| TOTALS (all libraries) | 26 |

1055 GETS were required to define 26 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:MMG/OBJ=OBJ\$:MMG MSRC\$:MMG/UPDATE=(ENH\$:MMG)+EXECML\$/LIB+LIB\$:SDALIB/LIB

0352

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY